REPORT DOCUMENTATION PAGE

AFRL-SR-BL-TR-01-

0032

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. I
		1 December 1992 to 30 November 1997

4. TITLE AND SUBTITLE
Automatic Methods and Tools for the Verification of Real Time Systems

5. FUNDING NUMBERS
F49620-93-1-0056

6. AUTHOR(S)
Thomas A. Henzinger

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Electrical Engineering and Computer Sciences
University of California
Berkeley, CA 94720-1770

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR
801 N. Randolph Street, Room 732
Arlington, VA 22203-1977

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
F49620-93-1-0056

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

We developed formal methods and tools for the verification of real-time systems. This was accomplished by extending techniques, based on automata theory and temporal logic, that have been successful for the verification of time-independent reactive systems. As system specification lan ~ maage for embedded real-time systems, we introduced hybrid automata, which equip traditional discrete automata with real-numbered clock variables and continuous environment variables. As requirements specification languages, we introduced temporal logics with clock variables for expressing timing constraints. Since the state spaces of systems with real-numbered clock variables are infinite, all verification must proceed symbolically. Symbolic verification methods are based either on deductive reasoning, using proof rules for symbolic logics, or on algorithmic analysis, using model checking procedures that operate on symbolic representations of state sets. We developed proof calculi for checking if a hybrid automaton satisfies linear-time clock properties, and we developed and implemented symbolic procedures for checking if a piecewise-linear hybrid automaton satisfies branching-time clock properties. The continuous variables of piecewise linear hybrid automata follow trajectories within piecewise-linear envelopes, which can be used to approximate conservatively the behavior of more general, nonlinear systems. We also studied the complexity of various formulations of the verification problem for real-time systems, and we identified the exact boundary between decidability and undecidability of real-time reasoning.

14. SUBJECT TERMS

15. NUMBER OF PAGES
8

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

Automatic Methods and Tools for the Verification of Real-time Systems
AFOSR Grant F49620-93-1-0056

Final Report
1 Dec 92 - 30 ~~Sep~~ Nov 97

Thomas A. Henzinger
Electrical Engineering and Computer Sciences
University of California
Berkeley, CA 94720-1770

Phone: (510) 643-2430
Fax: (510) 643-5052
Email: tah@eecs.berkeley.edu

Summary of Completed Project

We developed formal methods and tools for the verification of real-time systems. This was accomplished by extending techniques, based on automata theory and temporal logic, that have been successful for the verification of time-independent reactive systems. As system specification language for embedded real-time systems, we introduced hybrid automata, which equip traditional discrete automata with real-numbered clock variables and continuous environment variables. As requirements specification languages, we introduced temporal logics with clock variables for expressing timing constraints. Since the state spaces of systems with real-numbered clock variables are infinite, all verification must proceed symbolically. Symbolic verification methods are based either on deductive reasoning, using proof rules for symbolic logics, or on algorithmic analysis, using model checking procedures that operate on symbolic representations of state sets. We developed proof calculi for checking if a hybrid automaton satisfies linear-time clock properties, and we developed and implemented symbolic procedures for checking if a piecewise-linear hybrid automaton satisfies branching-time clock properties. The continuous variables of piecewise linear hybrid automata follow trajectories within piecewise-linear envelopes, which can be used to approximate conservatively the behavior of more general, nonlinear systems. We also studied the complexity of various formulations of the verification problem for real-time systems, and we identified the exact boundary between decidability and undecidability of real-time reasoning.

Deliverables

1. PhD Dissertations

Pei-Hsin Ho (1995): Automatic Analysis of Hybrid Systems
Peter W. Kopke (1996): The Theory of Rectangular Hybrid Automata

2. Publications

See http://www.eecs.berkeley.edu/~tah/Publications

3. Software

We developed and implemented HyTech, a symbolic model checker for the automatic analysis of embedded real-time systems. HyTech, together with usage information, can be downloaded from http://www.eecs.berkeley.edu/~tah/HyTech

Automatic Methods and Tools for the Verification of Real-time Systems
AFOSR Grant F49620-93-1-0056

Progress Report
1 Aug 96 - 31 July 97

Thomas A. Henzinger
Electrical Engineering and Computer Sciences
University of California
Berkeley, CA 94720-1770

Phone: (510) 643-2430
Fax: (510) 643-5052
Email: tah@eecs.berkeley.edu

OBJECTIVES

There has been no change in the objectives, which are:

1. Extending the model-checking methodology to the analysis of real-time and hybrid systems.

2. Building a prototype model-checking tool for the analysis of real-time and hybrid systems.

STATUS OF EFFORT

Over the last year, we obtained four significant new theoretical results, all of which are in the process of being implemented in HyTech and its successor, Mocha. First, we showed that for controllers that sample the system state, rather than watch it continuously, the control problem is solvable efficiently for a wider class of hybrid systems. Second, we solved the receptiveness problem for timed and hybrid systems, which is necessary for modular design and analysis. Third, we identified a class of temporal properties of system modules that can be model checked efficiently in isolation, without considering the complete system. Fourth, we developed an efficient algorithm for the hierarchical verification of reactive systems with fairness constraints, such as the progress of time.

ACCOMPLISHMENTS/NEW FINDINGS

We describe our four main results in greater detail.

Discrete-time control for rectangular hybrid automata [see 17 below]

Rectangular hybrid automata model digital control programs of analog plant environments. We study rectangular hybrid automata where the plant state evolves continuously in real-numbered time, and the controller samples the plant state and changes the control state discretely, only at the integer points in time. We prove that rectangular hybrid automata have finite bisimilarity quotients when all control transitions happen at integer times, even if the constraints on the derivatives of the variables vary between control states. This is sharply in contrast with the conventional model where control transitions may happen at any real time, and already the reachability problem is undecidable. Based on the finite

bisimilarity quotients, we give an exponential algorithm for the symbolic sampling-controller synthesis of rectangular automata. We show our algorithm to be optimal by proving the problem to be EXPTIME-hard. We also show that rectangular automata form a maximal class of systems for which the sampling-controller synthesis problem can be solved algorithmically.

Modularity for timed and hybrid systems [16]

In a trace-based world, the modular specification, verification, and control of live systems require each module to be receptive; that is, each module must be able to meet its liveness assumptions no matter how the other modules behave. For example, physical realizability, assume- guarantee reasoning about live trace inclusion, and controller synthesis for live trace inclusion all depend on the receptiveness condition. In a real-time world, liveness is automatically present in the form of diverging time. The receptiveness condition, then, translates to the requirement that a module must be able to let time diverge no matter how the environment behaves. We study the receptiveness condition for real-time systems by extending the model of Reactive Modules to timed and hybrid modules. We define the receptiveness of such a module as the existence of a winning strategy in a game of the module against its environment. By solving the game on region graphs, we present an (optimal) EXPTIME algorithm for checking the receptiveness of propositional timed modules. By giving a fixpoint characterization of the game, we present a symbolic procedure for checking the receptiveness of linear hybrid modules. Finally, we present an assume-guarantee principle for reasoning about timed and hybrid modules, and a method for synthesizing receptive controllers of timed and hybrid modules.

Alternating-time temporal logic [18]

Temporal logic comes in two varieties: linear-time temporal logic assumes implicit universal quantification over all paths that are generated by system moves; branching-time temporal logic allows explicit existential and universal quantification over all paths. We introduce a third, more general variety of temporal logic: alternating-time temporal logic offers selective quantification over those paths that are possible outcomes of games, such as the game in which the system and the environment alternate moves. While linear-time and branching-time logics are natural specification languages for closed systems, alternating-time logics are natural specification languages for open systems. For example, by preceding the temporal operator "eventually" with a selective path quantifier, we can specify that in the game between the system and the environment, the system has a strategy to reach a certain state. Also the problems of receptiveness, realizability, and controllability can be formulated as model-checking problems for alternating-time formulas.

Depending on whether we admit arbitrary nesting of selective path quantifiers and temporal operators, we obtain the two alternating-time temporal logics ATL and ATLstar. We interpret the formulas of ATL and ATLstar with respect to two models of composition for open systems, synchronous and asynchronous. In the case of synchronous ATL, the expressive power beyond CTL comes at no cost: the model-checking complexity of ATL is, for synchronous systems, linear in the size of the system and the length of the formula, and for asynchronous systems, quadratic. The symbolic model-checking algorithm for CTL extends with few modifications to synchronous ATL, and with more work, also to asynchronous ATL. This makes ATL an obvious candidate for the automatic verification of open systems. In the case of ATLstar, the model-checking problem is closely related to the synthesis problem for linear-time formulas, and requires doubly exponential time for both synchronous and asynchronous systems.

Fair simulation [15]

The simulation preorder for labeled transition systems is defined locally as a game that relates states with their immediate successor states. Liveness assumptions about transition systems are typically modeled using fairness constraints. Existing notions of simulation for fair transition systems, however, are not local, and as a result, many appealing properties of the simulation preorder are lost. We extend the local definition of simulation to account for fairness: system S fairly simulates system I iff in the simulation game, there is a strategy that matches with each fair computation of I a fair computation of S. Our definition enjoys a fully abstract semantics and has a logical characterization: S fairly simulates I iff every fair computation tree embedded in the unrolling of I can be embedded also in the unrolling of S or, equivalently, iff every Fair-AFMC formula satisfied by I is satisfied also by S (AFMC is the universal fragment of the alternation-free mu-calculus). The locality of the definition leads us to a polynomial-time algorithm for checking fair simulation for finite-state systems with weak and strong fairness constraints. Finally, fair simulation implies fair trace-containment, and is therefore useful as an efficiently-computable local criterion for proving linear-time abstraction hierarchies.

PERSONNEL SUPPORTED

* Faculty

Thomas A. Henzinger

* Post-Docs

Orna Kupferman

* Graduate Students

Peter Kopke (graduated August 1996)
Freddy Mang
Sriram Rajamani

* Visitors

Rajeev Alur

PUBLICATIONS

All papers can be found at http://www.eecs.berkeley.edu/~tah.

* ACCEPTED

* Books/Book Chapters

1 Thomas A. Henzinger, Howard Wong-Toi, Using HyTech to synthesize control parameters for a steam boiler, Formal Methods for Industrial Applications: Specifying and Programming the Steam Boiler Control (J.-R. Abrial, E. Borger, H. Langmaack, eds.), Lecture Notes in Computer Science 1165, Springer-Verlag, 1996, pages 265-282.

2 Rectangular Hybrid Automata, Verification of Digital and Hybrid Systems (K. Inan, ed.), NATO-ASI Series, Springer-Verlag, to appear.

* Journals

3 Thomas A. Henzinger, Peter W. Kopke, State equivalences for rectangular hybrid automata, Theoretical Computer Science, Special issue for CONCUR 96, to appear.

4 Thomas A. Henzinger, Orna Kupferman, Moshe Y. Vardi, A space-efficient on-the-fly algorithm for real-time model checking, Theoretical Computer Science, Special issue for CONCUR 96, to appear.

5 Thomas A. Henzinger, Pei-Hsin Ho, Howard Wong-Toi, HyTech: a model checker for hybrid systems, Software Tools for Technology Transfer, to appear.

6 Rajeev Alur, Thomas A. Henzinger, Peter W. Kopke, Real-time system = discrete system + clock variables, Software Tools for Technology Transfer, to appear.

7 Thomas A. Henzinger, Pei-Hsin Ho, Howard Wong-Toi, Algorithmic analysis of nonlinear hybrid systems, IEEE Transactions on Automatic Control, to appear.

8 Thomas A. Henzinger, Some Myths about Formal Verification, ACM Computing Surveys 28(A), 1996, http://www.acm.org/surveys/1996/HenzingerMyths/HenzingerMyths.html.

* Conferences

9 Thomas A. Henzinger, Peter W. Kopke, State equivalences for rectangular hybrid automata, Proceedings of the Seventh International Conference on Concurrency Theory (CONCUR 96), Lecture Notes in Computer Science 1119, Springer-Verlag, 1996, pages 530-545.

10 Thomas A. Henzinger, Orna Kupferman, Moshe Y. Vardi, A space-efficient on-the-fly algorithm for real-time model checking, Proceedings of the Seventh International Conference on Concurrency Theory (CONCUR 96), Lecture Notes in Computer Science 1119, Springer-Verlag, 1996, pages 514-529.

11 Thomas A. Henzinger, Orna Kupferman, From quantity to quality, Proceedings of the First International Workshop on Hybrid and Real-time Systems (HART 97), Lecture Notes in Computer Science 1201, Springer-Verlag, 1997, pages 48-62.

12 Vineet Gupta, Thomas A. Henzinger, Radha Jagadeesan, Robust timed automata, Proceedings of the First International Workshop on Hybrid and Real-time Systems (HART 97), Lecture Notes in Computer Science 1201, Springer-Verlag, 1997, pages 331-345.

13 Rajeev Alur, Robert K. Brayton, Thomas A. Henzinger, Shaz Qadeer, Sriram K. Rajamani, Partial-order reduction in symbolic state-space exploration, Proceedings of the 9th International Conference on Computer-aided Verification (CAV 97), Lecture Notes in Computer Science, Springer-Verlag, 1997, pages 340-351.

14 Thomas A. Henzinger, Pei-Hsin Ho, Howard Wong-Toi, HyTech: a model checker for hybrid systems, Proceedings of the 9th International Conference on Computer-aided Verification (CAV 97), Lecture Notes in Computer Science, Springer-Verlag, 1997, pages 460-463.

15 Thomas A. Henzinger, Orna Kupferman, Sriram K.~Rajamani, Fair simulation, Proceedings of the Eighth International Conference on Concurrency Theory (CONCUR 97), Lecture Notes in Computer Science 1243, Springer-Verlag, 1997, pages 273-287. Invited to a special issue of Fundamenta Informaticae.

16 Rajeev Alur, Thomas A.Henzinger, Modularity for timed and hybrid systems, Proceedings of the Eighth International Conference on Concurrency Theory (CONCUR 97), Lecture Notes in Computer Science 1243, Springer-Verlag, 1997, pages 74-88.

17 Thomas A. Henzinger, Peter W. Kopke, Discrete-time control for rectangular hybrid automata, Proceedings of the 24th International Colloquium on Automata, Languages, and Programming (ICALP 97), Lecture Notes in Computer Science 1256, Springer-Verlag, 1997, pages 582-593. Invited to a special issue of Theoretical Computer Science.

18 Rajeev Alur, Thomas A. Henzinger, Orna Kupferman, Alternating-time temporal logic, Proceedings of the 38th Annual IEEE Symposium on Foundations of Computer Science (FOCS 97), to appear.

19 Rajeev Alur, Thomas A. Henzinger, Symbolic analysis of hybrid systems, Proceedings of the Annual IEEE Conference on Decision and Control (CDC 97), to appear.

INTERACTIONS/TRANSITIONS

* Participation/Presentations At Meetings, Conferences, Seminars, Etc.

Invited lectures of the PI

Rectangular Hybrid Automata, NATO-ASI International Summer School on the Verification of Digital and Hybrid Systems, Antalya, Turkey (June 1997).

Models and Logics for Timed and Hybrid Systems: An Introduction, School on Methods and Tools for the Verification of Infinite-state Systems, Grenoble, France (March 1997).

Some Lessons from the {\sc HyTech} Experience, School on Methods and Tools for the Verification of Infinite-state Systems, Grenoble, France (March 1997).

Automatic Verification of Real-time and Hybrid Systems, BRICS Autumn School on Verification, Aarhus, Denmark (October 1996).

A Brief History of Real Time, First International Workshop on the Verification of Infinite-state Systems, Pisa, Italy (August 1996).

Other lectures of the PI

Discrete-time control for rectangular hybrid automata, Cadence European Labs, Rome, Italy (July 1997).

Computer-aided Verification of Embedded Systems, Computer Science Colloquium, University of British Columbia, Vancouver, Canada (January 1997); Max-Planck-Institut f"ur Informatik, Saarbr"ucken, Germany (March 1997).

Formal Verification of Embedded Systems, Industrial Liaison Program, University of California, Berkeley, California (March 1997).

Efficient Verification of Heterogeneous Systems, SRC Formal Verification Review, Carnegie-Mellon University, Pittsburgh, Pennsylvania (March 1997).

Efficient Formal Verification using Transition Hierarchies, Intel Development Labs, Hillsboro, Oregon (January 1997).

A Brief History of Real Time, University of Oldenburg, Oldenburg, Germany (October 1996); SRI International, Menlo Park, California (November 1996).

Algorithmic Analysis of Real-time and Hybrid Systems, University of Passau, Passau, Germany (September 1996).

Other professional activities of the PI

Organizer and program co-chair, Eighth International Conference on Computer-aided Verification (CAV 96), New Brunswick, New Jersey (July 1996).

Editor (with Rajeev Alur), Computer-aided Verification: Proceedings of the Eighth International Conference, Lecture Notes in Computer Science 1102, Springer-Verlag, 1996.

Panelist, Future Trends in Industrial Computer-aided Verification, Ninth International Conference on Computer-aided Verification (CAV 97), Haifa, Israel (June 1997).

Editor, Formal Methods in System Design.

Editor, Software Tools for Technology Transfer.

Program committee member, Seventh International Conference on Concurrency Theory (CONCUR 96), Pisa, Italy (August 1996).

Program committee member, Fifth International Hybrid Systems Workshop, Notre Dame, Indiana (September 1997).

Program committee member, International Symposium on Theoretical Aspects of Computer Software (TACS 97), Sendai, Japan (September 1997).

Program committee member, Second International Workshop on the Verification of Infinite-state Systems (Infinity 97), Bologna, Italy (July 1997).

Program committee member, Ninth International Conference on Computer-aided Verification (CAV 97), Haifa, Israel (June 1996).

Program committee member, Fourth AMAST Workshop on Real-Time Systems, Concurrent, and Distributed Software (ARTS 97), Mallorca, Spain (May 1996).

Program committee member, Formal Aspects of Software Engineering (FASE 97), Lille, France (April 1997).

Program committee member, International Workshop on Hybrid and Real-Time Systems (HART 97), Grenoble, France (March 97).

Program committee member, First ACM SigPlan Workshop on the Automated Analysis of Software, Paris, France (January 1997).

Program committee member, Fourth International Symposium on Formal Techniques in Real-time and Fault-tolerant Systems (FTRTFT 96), Uppsala, Sweden (September 1996).

Program committee member, Fourth International Hybrid Systems Workshop, Ithaca, New York (October 1996).

* Consultative And Advisory Functions To Other Laboratories And Agencies

Consultant, DARPA 1997 ISAT study group on Complex Systems.

* Transitions

The current version of HyTech, Version 1.04a, was released in May 1997. HyTech is publicly available on the world-wide web at http://www.eecs.berkeley.edu/~tah/HyTech.

NEW DISCOVERIES, INVENTIONS, OR PATENT DISCLOSURES

None.

HONORS/AWARDS

None.